CoRAG: Collaborative Retrieval-Augmented Generation

Aashiq Muhamed[1], Mona Diab[1], Virginia Smith[2]
{amuhamed, mdiab, smithv}@andrew.cmu.edu
[1] Language Technologies Institute, [2] Machine Learning Department
Carnegie Mellon University

Abstract

Retrieval-Augmented Generation (RAG) models excel in knowledge-intensive tasks, especially under few-shot learning constraints. We introduce CoRAG, a framework extending RAG to collaborative settings, where clients jointly train a shared model using a collaborative passage store. To evaluate CoRAG, we introduce CRAB, a benchmark for collaborative homogeneous open-domain question answering. Our experiments demonstrate that CoRAG consistently outperforms both parametric collaborative learning methods and locally trained RAG models in low-resource scenarios. Further analysis reveals the critical importance of relevant passages within the shared store, the surprising benefits of incorporating irrelevant passages, and the potential for hard negatives to negatively impact performance. This introduces a novel consideration in collaborative RAG: the trade-off between leveraging a collectively enriched knowledge base and the potential risk of incorporating detrimental passages from other clients. Our findings underscore the viability of CoRAG, while also highlighting key design challenges and promising avenues for future research[1].

1 Introduction

Retrieval-Augmented Generation (RAG) models (Lewis et al., 2020; Izacard et al., 2022; Qin et al., 2019; Zhang et al., 2021), which incorporate large external datastores of text passages, have shown promise in knowledge-intensive and few-shot tasks. However, their exploration has mainly focused on centralized settings where a single entity controls both the model and the datastore. The potential of RAG within a collaborative learning framework, where multiple clients jointly train a shared model without directly exchanging their labeled data (McMahan et al., 2016), but potentially building a shared passage store, remains largely unexplored. Consider competing businesses in the same industry, each possessing expensive to acquire (labeled) data on customer behavior. Directly sharing these data would be strategically disadvantageous, yet they could collaborate to build a shared passage store of relatively inexpensive (unlabeled) market research documents and economic analyses. This allows them to collectively train a more effective RAG model for market prediction without revealing their valuable labeled data. This approach, particularly in low-resource settings enables them to train a more effective model than any single client could achieve independently.

This work introduces CoRAG, a framework for collaborative RAG that enables multiple clients to jointly train a shared model using a collaborative passage store, while allowing them to use their local passage stores during inference. CoRAG introduces unique challenges stemming from the dynamics of constructing and utilizing this shared store. The composition of this knowledge base, particularly the balance of relevant, irrelevant, and hard-negative passages, significantly impacts the model's performance and generalization capabilities. Our experiments reveal that relevant passages are crucial for model generalization, while hard negatives can be detrimental, and, surprisingly, irrelevant passages can even be beneficial. This introduces a fundamental tension in CoRAG: clients must balance the advantages of a richer, shared knowledge base with the risk of incorporating potentially detrimental passages from others. To explore these dynamics, we introduce CRAB, a homogeneous open-domain question answering benchmark. Using CRAB, we empirically demonstrate that a carefully curated collaborative store, rich in relevant passages and minimizing hard negatives, significantly improves model performance compared to parametric collaborative learning methods and local RAG training. Our contributions include:

[1]Code is available at https://github.com/aashiqmuhamed/CoRAG

Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers), pages 265–276
April 29 - May 4, 2025 ©2025 Association for Computational Linguistics

- **CoRAG Framework:** We introduce CoRAG, a framework for collaborative training of RAG models. CoRAG enables multiple clients to jointly train a shared model using a collaborative passage store, while allowing the use of client-specific stores during inference. We show that using a collaborative passage store can significantly improve few-shot performance over collaborative parametric or local RAG models.
- **Passage Composition and Client Incentives:** We investigate how the composition of the collaborative store (relevant, irrelevant, and hard-negative passages) affects model generalization and client participation incentives. Our analysis uncovers a fundamental tension: clients must weigh the benefits of accessing an enriched collaborative store against the risk of incorporating potentially detrimental passages from other clients.

Algorithm 1 Collaborative Retrieval-Augmented Generation

Require: M clients, Pretraining data D_{pre}, Train question answer pairs per client $\{D_{\text{train},i}\}_{i=1}^{M}$, Collaborative train passage store I_{train}, Test passage stores $\{I_{\text{test},i}\}_{i=1}^{M}$, Test queries $\{Q_i\}_{i=1}^{M}$
Ensure: Responses $\{O_i\}_{i=1}^{M}$
Pretraining:
Pretrain retriever R and reader G using D_{pre}
Collaborative Training:
for each round **do**
  for each client i **do**
    $R_i, G_i \leftarrow R, G$ ▷ Init with global model
    $P_i \leftarrow R(D_{\text{train},i}, I_{\text{train}})$ ▷ Retrieve passages
    Update local R_i, G_i using P_i and $D_{\text{train},i}$
  end for
  $R, G \leftarrow \text{Aggregate}(\{R_i, G_i\}_{i=1}^{M})$ ▷ Update global model
end for
Inference:
for each client i **do**
  $P_i \leftarrow R(Q_i, I_{\text{test},i})$ ▷ Retrieve client i passages
  $O_i \leftarrow G(Q_i, P_i)$ ▷ Generate client i response
end for
return $\{O_i\}_{i=1}^{M}$

2 CoRAG Framework

RAG models (Lewis et al., 2020; Izacard et al., 2022) enhance parametric LMs by incorporating external knowledge in the form of a passage store. Given an input x (e.g., a question), a RAG model retrieves relevant documents z from the passage store and uses them to generate an output y (e.g., an answer). The model estimates the probability of generating y given x, denoted as $p_{RAG}(y|x)$, by marginalizing over the top k retrieved documents:

$$p_{\text{RAG}}(y|x) \approx \sum_{z \in \text{top-}k(R(\cdot|x))} R(z|x) \prod_{i=1}^{N} G(y_i|z, x, y_{1:i-1})$$

CoRAG (Algorithm 1) combines collaborative learning with RAG models, enabling clients to jointly train a shared model while leveraging a collaboratively constructed passage store. This is particularly advantageous in low-resource settings, where individual clients may have limited local data. By pooling their knowledge through a shared passage store, clients gain access to a broader and more diverse knowledge base, facilitating improved learning and generalization.

CoRAG operates in three phases: During *Pretraining*, each retriever and reader are pretrained on a large, shared dataset D_{pre} using self-supervised objectives to enable general language understanding. In the *Collaborative Learning* phase, clients collaboratively finetune the pretrained retriever and reader on their local training datasets $\{D_{\text{train},i}\}_{i=1}^{M}$ by retrieving relevant passages from a collaborative passage store I_{train}, constructed through contributions from all participating clients. Client model updates are aggregated in a decentralized or centralized fashion (e.g., using a method such as FedAvg (McMahan et al., 2016)), producing a global model that reflects the collective knowledge gained during collaborative training. In the *Inference* phase, clients utilize the collaboratively trained global RAG model to process incoming queries. Each client aims to maximize local question-answering metrics by identifying relevant passages from a local test passage store I_{test} that may include passages from the collaborative index and new client-specific passages.

In addition to the Reader and Retriever, CoRAG employs the Collaborative Passage Store I_{train}, a collection of text passages contributed by all participating clients. Separate passage stores are used for training and testing, with their composition (relevant, irrelevant, and hard-negative passages) significantly influencing both model performance and client incentives for contributing high-quality passages, as we will explore further.

3 Experiments and Results

3.1 CRAB: Collaborative RAG Benchmark

To investigate passage composition in CoRAG, we introduce CRAB, a homogeneous (identically distributed across clients) open-domain QA benchmark derived from NaturalQuestions (Kwiatkowski et al., 2019) with train, test, and

dev splits distributed across 8 clients. To study few-shot learning, we provide train splits with 16, 32, and 64 sampled training QA pairs per client. The unique dev (8752 pairs) and test QA pairs (3600 pairs) are evenly split among clients.

The passage datastore for CRAB is derived from the Wikipedia 32M passages (wiki-dec2018) (Izacard et al., 2022). Mirroring real-world scenarios where new documents emerge or shared knowledge becomes inaccessible, CRAB incorporates distinct passage stores for training and testing, ensuring no overlapping passages between them. While test and dev passages are unique to each client, overlaps in relevant passages are possible between different clients. We will release passage stores corresponding to the various passage composition experiments in this work.

3.2 Experimental Setup

CoRAG is instantiated with Contriever (Izacard et al., 2021) as the retriever and a pretrained T5 base model with Fusion-in-Decoder (Izacard and Grave, 2020) as reader on all 8 clients. We compare its performance against flan-t5-base (Chung et al., 2022), a comparable-sized (∼220M parameters) closed-book (no retrieval) instruction-tuned parametric model. We focus on smaller models as they are more practical in resource-constrained collaborative learning settings, where communication overhead can be a significant limitation (Woisetschläger et al., 2024; Nguyen et al., 2022). We pretrained all models on 350 million passages from 2021 Wikipedia and a subset of the 2020 Common Crawl (Thurner et al., 2018). They are then finetuned using bloat16 precision using FedAvg on CRAB in few-shot settings (16, 32, and 64 training examples per client). We use the Perplexity Distillation loss (Izacard et al., 2023) for both pretraining and finetuning. We report the best client-averaged Exact match score (EM) on the test set across rounds, and the micro-averaged metrics for the Centralized baseline.

We employ the AdamW optimizer with a batch size of 64 and a learning rate of 4×10^{-5} with linear decay for both the reader and retriever. The retriever is trained using query-side finetuning. We employ greedy decoding to generate the answers. During both training and testing, we retrieve the top 40 passages and truncate the concatenation of the query and the retrieved passages to a maximum of 384 tokens. For *Collaborative Training*, we do not use warmup iterations, train for 10 rounds with 64 epochs per round, and evaluate the model at the end of each round. For *Local Training*, we use 20 warmup iterations, train for 1000 steps, and evaluate the model every 100 steps. All models were trained on 4 A6000 GPUs in under a day. Further details are in Appendix B.

3.3 CoRAG is Effective in Few-shot Settings

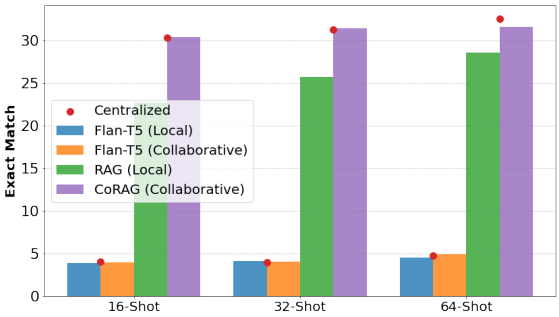

Figure 1: Performance of Flan-T5, RAG (Local), and CoRAG on CRAB. CoRAG consistently outperforms Flan-T5 across training configurations. Performance gap between CoRAG and baselines widens as training samples per client decreases.

Fig 1 compares the few-shot performance of CoRAG against RAG (Local) model and Flan-T5 on CRAB. CoRAG leverages a shared passage store containing the entire Wikipedia, RAG (Local) uses an evenly partitioned Wikipedia across clients to simulate real-world settings, while Flan-T5 relies solely on its parametric knowledge. We evaluate all models in Centralized (combining datasets from all clients), Local (individual client train sets), and Collaborative (locally trained, aggregated after each round) configurations.

We find that (i) CoRAG (Collaborative) and RAG (Local) consistently surpass the parametric-only baseline (Flan-T5) in collaborative and local training configurations respectively, across shot settings. (ii) Leveraging the shared passage store confers an advantage to CoRAG over local training. (iii) CoRAG proves particularly effective under limited labeled Q/A pairs per client, showing a 10.5% improvement over RAG (Local) at 64-shot, which increases to 33.8% at 16-shot. (iv) CoRAG performance is close to Centralized, consistent with previous observations in benchmarks with homogeneous (identically distributed) client data. These results establish CoRAG as a promising direction for few-shot learning.

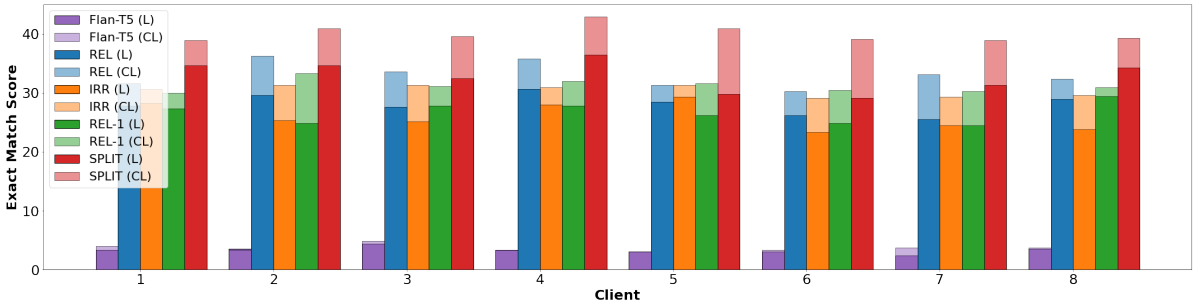

Figure 2: 64-shot EM scores on the CRAB benchmark. L is Local and CL is Collaborative. CoRAG consistently improves over RAG (Local) across all clients (1-8) and store choices. Improvement varies depending on the composition of passage store.

3.4 Impact of Passage Store Composition

We investigate how the *train* passage store composition impacts few-shot QA performance. We classify the BM25-retrieved passages for each concatenated QA pair as a query. The passages are categorized as relevant (top-5 passages containing the ground truth answer), hard negatives (ranked 6–50), and irrelevant (all remaining passages). To validate our categorization, we manually inspected 100 question-answer pairs and confirmed that our chosen ranges effectively captured the intended distinctions. We construct four train passage stores: (1) REL: Collaborative store containing relevant passages for all client QA data + 80% of Wikipedia (2) IRR: Collaborative store containing 80% of Wikipedia, but excluding all relevant passages (3) REL-1: Seven clients use IRR; one client uses IRR + relevant passages for all client QA data (4) SPLIT: Each client store has relevant passages for their own QA data + 10% of Wikipedia. The disjoint test sets I_{test} are client-local and comprise relevant passages for the test QA data and 2.5% of Wikipedia.

Table 1 compares the 64-shot performance of RAG (Local) and CoRAG on the four store variants. CoRAG consistently outperforms RAG (Local) across all train store variants, and matches the Centralized RAG baseline. The presence of relevant passages in REL significantly improves performance over IRR, confirming their importance for generalization. Interestingly, concentrating relevant passages in a single client (REL-1) only marginally improves over IRR. This is because the benefits manifest through indirect information flow: relevant passages improve client 8's generalization (see Figure 2), which then propagates to other clients via collaborative training. Finally, SPLIT, with a higher concentration of client-specific relevant passages, further boosts performance, highlighting the benefits of selectively concentrating relevant passages during training.

Passage Store →	REL	IRR	REL-1	SPLIT
RAG (Local)	28.088	25.944	26.597	34.694
CoRAG	33.011	30.444	30.944	40.056

Table 1: Average EM under various passage store options. CoRAG outperforms RAG (Local). REL outperforms IRR, highlighting the importance of relevant passages. SPLIT outperforms REL, showing the benefit of passage concentration.

Table 2 analyzes how training passage store composition affects RAG (Local) performance. Randomly downsampling irrelevant and hard-negative passages from REL has minimal impact. Notably, including hard negatives during training generally decreases performance, while irrelevant passages tend to improve performance.

Our initial investigation suggests two possible mechanisms underlying these trends. First, from the retriever's perspective, hard negatives introduce ambiguity in non-contrastive RAG training, as their partial lexical and semantic overlap with gold passages generates weak or contradictory gradient signals. Unlike contrastively trained retrievers, which explicitly optimize for hard negative separation, the end-to-end RAG training framework lacks a structured push-away mechanism, leading to suboptimal passage ranking. In contrast, irrelevant passages act as easy negatives, creating a cleaner decision boundary between relevant and non-relevant documents, thereby reinforcing retriever robustness. Second, from the reader's perspective, irrelevant passages may mitigate entropy collapse, a failure mode in which excessively low attention entropy causes the model to overcommit to misleading context. This more diffuse distribution of attention ultimately improves test-time RAG performance (Cuconasu et al., 2024).

Train Passage Store Composition	Exact Match
Only relevant	29.111
Only hard neg + irrelevant	25.222
Only relevant + hard neg	25.778
Only relevant + irrelevant	**32.667**
Only top-1 relevant + irrelevant	31.556

Table 2: Effect of training passage store composition on RAG (local) test performance averaged across 8 clients. Hard negatives hurt performance, while irrelevant passages are surprisingly beneficial.

3.5 Client Incentives

We observe in Figure 2 that CoRAG outperforms RAG (Local) across all passage stores, with gains varying based on store composition. This introduces a novel challenge in CoRAG: strategically deciding which passages to contribute. Unlike traditional collaborative learning, CoRAG introduces a tension between maximizing individual utility and contributing to the collective knowledge base. Contributing high-quality passages benefits all clients but risks incorporating detrimental hard negatives from others. Clients with many relevant passages might be reluctant to contribute, fearing dilution of their advantage, while those with fewer relevant passages stand to gain more from collaboration.

The decision to contribute balances potential improvements from accessing a larger passage pool against the risk of incorporating hard negatives. Appendix G formalizes this trade-off in a client utility model. Addressing this tension requires designing mechanisms that incentivize high-quality contributions while ensuring equitable participation, such as contribution-based rewards, tiered access levels, and reputation systems to track client contribution history.

4 Conclusion and Future Work

This work introduces CoRAG, a framework extending RAG to collaborative learning, enabling clients to jointly train a shared model and collaboratively construct a passage store. Our experiments on CRAB, a collaborative QA benchmark, demonstrate the significant performance advantage of CoRAG in few-shot settings. We analyze the impact of passage store composition on performance, highlighting the importance of relevant and, surprisingly, irrelevant passages, while showing the detrimental effects of hard negatives. Future work includes evaluating CoRAG on heterogeneous client distributions, and designing robust incentive mechanisms.

Acknowledgements

This work was supported in part by the National Science Foundation grants IIS2145670 and CCF2107024, and funding from Amazon, Apple, Google, Intel, Meta, and the CyLab Security and Privacy Institute. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of any of these funding agencies.

5 Limitations

Our work presents a promising step towards collaborative RAG, but it is important to acknowledge its limitations and highlight areas for future research.

Homogeneous Data Distribution. Our experiments focus on a homogeneous setting where clients have identically distributed data. This simplification allows us to isolate the impact of passage composition and client incentives. However, real-world collaborative scenarios often involve heterogeneous data distributions, where clients possess data from different sources, domains, or with varying levels of quality. Evaluating CoRAG's effectiveness and fairness under heterogeneous settings is am important area for future work.

Scalability and Efficiency. Our experiments are conducted on a relatively small scale with 8 clients. Scaling CoRAG to a larger number of clients, potentially with diverse computational resources and communication constraints, presents challenges related to communication efficiency, model aggregation, and handling of large passage stores. Exploring optimization strategies to enhance scalability is a promising direction for future research.

Incentive Mechanism Design. We propose potential incentive mechanisms to address the tension between individual utility and contributing to the common good. However, designing, evaluating, and deploying robust incentive mechanisms that effectively promote high-quality contributions while ensuring fairness requires further investigation.

6 Ethical Considerations

While CoRAG offers promising benefits for few-shot collaborative model training, we acknowledge and address the potential ethical considerations associated with its development and deployment.

Bias. The shared passage store, constructed collaboratively by multiple clients, may inadvertently reflect biases present in the data held by individual clients. This could lead to unfair or discriminatory outcomes, particularly if the trained model is used in applications that impact decision-making. Mitigating this risk requires developing robust mechanisms for bias detection and mitigation during the construction and maintenance of the shared store.

Misuse. The capabilities of CoRAG could be exploited for malicious purposes, such as generating harmful or misleading content. Safeguards against such misuse are essential and could include access control mechanisms, content moderation strategies, and clear ethical guidelines for using the technology.

Equity and Fairness. The benefits of collaborative RAG should be accessible to all participating clients, regardless of their data resources or technical capabilities. This requires designing incentive mechanisms that encourage contributions from a diverse range of clients and providing support to those with limited data or expertise to ensure equitable participation.

Addressing these ethical considerations throughout the design, development, and deployment of CoRAG systems can help ensure their responsible use.

Data & Licensing Considerations

To ensure reproducibility and facilitate further research in collaborative retrieval-augmented generation, we release the following resources under permissive licenses:

- **CoRAG Codebase:** The complete codebase for implementing CoRAG, including the retriever, reader, training procedures, and code for generating the different passage store variants.
- **CRAB Dataset:** The CRAB benchmark dataset, including the data splits, the passage datastore, and the evaluation scripts. This dataset is constructed using the NaturalQuestions dataset, which is released under the Apache License 2.0, and the Wikipedia 32M passages (wiki-dec2018) dataset, which is publicly available. Our use of these datasets is consistent with their intended use and licensing terms.

We have documented configurations, prompt details, training procedures, and hyperparameter selection in Appendix B, to ensure reproducibility. All publicly available datasets used in this work have followed accepted privacy practices at the time of their creation.

References

Yae Jee Cho, Divyansh Jhunjhunwala, Tian Li, Virginia Smith, and Gauri Joshi. 2022. Maximizing global model appeal in federated learning. *arXiv preprint arXiv:2205.14840*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv: 2210.11416*.

Florin Cuconasu, Giovanni Trappolini, Federico Siciliano, Simone Filice, Cesare Campagnano, Yoelle Maarek, Nicola Tonellotto, and Fabrizio Silvestri. 2024. The power of noise: Redefining retrieval for rag systems. *arXiv preprint arXiv: 2401.14887*.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The faiss library.

Masoomali Fatehkia, Ji Kim Lucas, and Sanjay Chawla. 2024. T-rag: Lessons from the llm trenches. *arXiv preprint arXiv: 2402.07483*.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Nika Haghtalab, Michael Jordan, and Eric Zhao. 2022. On-demand sampling: Learning optimally from multiple distributions. *Advances in Neural Information Processing Systems*, 35:406–419.

Zhiyuan He, Huiqiang Jiang, Zilong Wang, Yuqing Yang, Luna Qiu, and Lili Qiu. 2024. Position engineering: Boosting large language models through positional information manipulation. *arXiv preprint arXiv: 2404.11216*.

Baihe Huang, Sai Praneeth Karimireddy, and Michael I Jordan. 2023. Evaluating and incentivizing diverse data contributions in collaborative learning. *arXiv preprint arXiv:2306.05592*.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin,

and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *Trans. Mach. Learn. Res.*

Gautier Izacard and Edouard Grave. 2020. Leveraging passage retrieval with generative models for open domain question answering. *Conference of the European Chapter of the Association for Computational Linguistics*.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2023. Atlas: Few-shot learning with retrieval augmented language models. *Journal of Machine Learning Research*, 24(251):1–43.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2022. Few-shot learning with retrieval augmented language model. *arXiv preprint arXiv: Arxiv-2208.03299*.

Sai Praneeth Karimireddy, Wenshuo Guo, and Michael I. Jordan. 2022. Mechanisms that incentivize data sharing in federated learning. *arXiv preprint arXiv: 2207.04557*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

H. B. McMahan, Eider Moore, Daniel Ramage, S. Hampson, and B. A. Y. Arcas. 2016. Communication-efficient learning of deep networks from decentralized data. *International Conference on Artificial Intelligence and Statistics*.

Sewon Min, Suchin Gururangan, Eric Wallace, Hannaneh Hajishirzi, Noah A. Smith, and Luke Zettlemoyer. 2023. Silo language models: Isolating legal risk in a nonparametric datastore. *arXiv preprint arXiv: 2308.04430*.

John Nguyen, Jianyu Wang, Kshitiz Malik, Maziar Sanjabi, and Michael Rabbat. 2022. Where to begin? on the impact of pre-training and initialization in federated learning. *arXiv preprint arXiv:2206.15387*.

Marc Pickett, Jeremy Hartman, Ayan Kumar Bhowmick, Raquib ul Alam, and Aditya Vempaty. 2024. Better rag using relevant information gain. *arXiv preprint arXiv: 2407.12101*.

Lianhui Qin, Michel Galley, Chris Brockett, Xiaodong Liu, Xiang Gao, William B. Dolan, Yejin Choi, and Jianfeng Gao. 2019. Conversing by reading: Contentful neural conversation with on-demand machine reading. In *Annual Meeting of the Association for Computational Linguistics*.

Stefan Thurner, Rudolf Hanel, and Peter Klimekl. 2018. Scaling. *Oxford Scholarship Online*.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzm'an, Armand Joulin, and Edouard Grave. 2019. Ccnet: Extracting high quality monolingual datasets from web crawl data. *International Conference on Language Resources and Evaluation*.

Herbert Woisetschläger, Alexander Erben, Shiqiang Wang, Ruben Mayer, and Hans-Arno Jacobsen. 2024. Federated fine-tuning of llms on the very edge: The good, the bad, the ugly. In *Proceedings of the Eighth Workshop on Data Management for End-to-End Machine Learning*, DEEM '24, page 39–50, New York, NY, USA. Association for Computing Machinery.

Lukas Wutschitz, Boris Köpf, Andrew Paverd, Saravan Rajmohan, Ahmed Salem, Shruti Tople, Santiago Zanella-Béguelin, Menglin Xia, and Victor Rühle. 2023. Rethinking privacy in machine learning pipelines from an information flow control perspective. *arXiv preprint arXiv:2311.15792*.

Yizhe Zhang, Siqi Sun, Xiang Gao, Yuwei Fang, Chris Brockett, Michel Galley, Jianfeng Gao, and Bill Dolan. 2021. Joint retrieval and generation training for grounded text generation. *ArXiv*, abs/2105.06597.

A Related Work

Collaborative Learning. Collaborative learning (CL) (McMahan et al., 2016; Cho et al., 2022; Huang et al., 2023; Haghtalab et al., 2022; Karimireddy et al., 2022) enables multiple clients to jointly train a shared model without directly sharing their raw data. Traditional CL methods primarily focus on parametric models, where the shared model is represented by a set of parameters that are updated iteratively based on client contributions.

Retrieval-Augmented Generation. RAG models (Lewis et al., 2020; Izacard et al., 2022; Gao et al., 2023) augment parametric language models with a large external datastore of text passages, enabling them to access and utilize a richer knowledge base. Centralized RAG has shown impressive performance in various tasks, including few-shot learning, open-ended question answering, and knowledge-grounded generation.

Data-Centric RAG. Recent works have explored the impact of context composition on RAG performance at inference time (Cuconasu et al., 2024; Pickett et al., 2024; Fatehkia et al., 2024; He et al., 2024). For example, Cuconasu et al. (2024) demonstrated that incorporating irrelevant passages during inference can improve generalization. Our work investigates this phenomenon during *training* within a collaborative setting, studying the role of passage composition.

Privacy-Preserving RAG. Recent work has explored using RAG to enhance privacy and compliance in centralized settings. Min et al. (2023) proposed Silo-LM, a language model that trains a parametric component on low-risk data and uses a separate nonparametric datastore for high-risk data, only accessing the latter during inference. Wutschitz et al. (2023) investigated privacy in language modeling from an information flow control perspective, finding that RAG offers superior utility and scalability while maintaining perfect secrecy.
Our work builds upon existing work by:

- Introducing CoRAG, a novel framework for collaborative RAG that enables clients to jointly train a shared model and leverage a collaboratively constructed passage store.
- Systematically analyzing the data-centric aspects of collaborative RAG, focusing on the impact of passage composition on both model generalization and client incentives.
- Highlighting the unique challenges related to passage contribution in collaborative RAG and proposing potential directions for incentive mechanism design to address these challenges.

B Training Details and Hyperparameters

For question answering on the CRAB benchmark, we format the input using the following template:

```
question: {question text} answer: [MASK_0]
```

The model is then trained to generate the masked token followed by the answer:

```
[MASK_0] {answer}.
```

We employ greedy decoding to generate the answers. During both training and testing, we retrieve the top 40 passages and truncate the concatenation of the query and the retrieved passages to a maximum of 384 tokens.

Hyperparameter Settings. All models are trained using bfloat16 precision. For both the parametric baseline (Flan-T5-base) and CoRAG, we employ the AdamW optimizer with a batch size of 64 and a learning rate of 4×10^{-5} with linear decay for both the language model and the retriever. The retriever is trained using query-side fine-tuning.

Training Procedures. The training procedures for collaborative and local settings differ slightly. Unless otherwise specified, we report the average of three runs.
Collaborative Training: We do not use warmup iterations, train for 10 rounds with 64 epochs per round, and evaluate the model at the end of each round. For collaborative training, we utilize FedAvg (McMahan et al., 2016) for model aggregation at the server, and we train on 8 clients.
Local Training: We use 20 warmup iterations, train for 1000 steps, and evaluate the model every 100 steps.

Compute All models were trained on 4 A6000 GPUs in under a day. We use exact MIPS search using FAISS (Douze et al., 2024), and all indices can be constructed in under 8 hours on a single A6000.

C Pretraining Data

Both CoRAG and RAG (Local) retriever and reader are pretrained on a datastore consisting of 350 million passages from the 2021 Wikipedia dump and a subset of the 2020 Common Crawl

dump (Thurner et al., 2018). This pretraining aims to provide a strong foundation for general language understanding.

The parametric Flan-T5-base model used in our experiments was also pretrained on Common Crawl (Wenzek et al., 2019), which includes English Wikipedia. While this pretraining provides general language capabilities, these models generally do not perform well on open-domain question-answering benchmarks like NaturalQuestions without further fine-tuning. This is because the pretraining data and objectives are not specifically tailored for open-domain question answering.

D Few-Shot Performance on CRAB

Table 3 reports the performance of Flan-T5, T5-base, and RAG (Local and Collaborative) on the CRAB benchmark in few-shot settings.
Table 4 presents the corresponding performance on the CRAB development set.

E Impact of Passage Store Composition

To better understand the impact of passage store composition on local RAG performance, we evaluated the client model's performance after adjusting the composition of the REL passage store I_{train} in Table 5. Recall that the REL store contains all relevant passages for the training data. In addition to the results in subsection 3.4, this table presents results where the relevant passages are kept constant, while the irrelevant and hard-negative passages are uniformly subsampled. This subsampling, which maintains the original proportion of hard negatives to irrelevant passages, has minimal impact on performance. We also observe that removing relevant passages during training is less detrimental than removing them during inference, as the test passage store always contains relevant passages.

Our analysis reveals a nuanced impact of passage store composition on local RAG performance. Incorporating hard negatives into the collaborative store generally leads to lower Exact Match and F1 scores. This suggests that hard negatives, despite their similarity to relevant passages, can mislead the retriever during training, leading to reduced performance at inference time. This differs from the findings in the contrastive learning literature, where hard negatives can be beneficial. In general, the composition of collaborative passages during training can affect test-time performance in several ways: (1) Distribution Shift: there is a shift between the collaborative passage store used during training and the client-specific passage stores used at inference. (2) Retriever Generalization: improving the training composition can enhance the retriever's ability to identify relevant passages at test time. (3) Reader Utilization: a better training composition can also improve the reader's ability to utilize those retrieved passages effectively. However, as CoRAG fine-tuning is not contrastive, it treats all retrieved passages equally, leading to reduced performance when hard negatives similar to relevant passages are present during training. However, including irrelevant passages in the collaborative store that are easier to distinguish often improves performance, indicating their potential role in helping the retriever learn to discriminate between relevant and irrelevant information.

F Client-Specific Performance Gains on CRAB

Table 6 presents the per-client performance gain of CoRAG over RAG (Local) for the various passage store configurations in the CRAB benchmark. This data was used to generate Figure 2, which visually depicts the impact of collaboration on individual client performance.

G Formalizing Client Incentives

The collaborative nature of CoRAG introduces a novel tension between maximizing individual utility and contributing to the collective knowledge base. Unlike traditional collaborative learning, CoRAG requires clients to strategically decide which passages to contribute, balancing potential improvements from accessing a larger passage pool against the risk of incorporating hard negatives from other clients.

Definitions and Notation Let N be the number of clients. For each client $i \in [N]$, we define:

- D_i: The local training data of client i.
- P_i: The set of all passages available to client i.
- R_i: The set of all passages relevant to client i's training data D_i. Note that R_i is not necessarily a subset of P_i.
- HN_i: The set of all hard negative passages for client i. These are passages that appear relevant to client i's retriever but do not contain the correct answer for D_i.
- IR_i: The set of all irrelevant passages for client i, i.e., passages that are neither in R_i nor in HN_i.

	T5-base		Flan-T5-base		RAG	
	EM ↑	F1 ↑	EM ↑	F1 ↑	EM ↑	F1 ↑
Centralized (64-shot)	3.340	6.892	4.810	8.678	32.556	41.071
Local (64-shot)	3.084	6.531	4.584	8.350	28.639	36.178
Collaborative (64-shot)	3.627	7.199	4.944	8.770	31.639	39.900
Centralized (32-shot)	2.880	6.292	4.011	7.933	31.324	39.250
Local (32-shot)	2.572	5.938	4.138	8.175	25.722	33.630
Collaborative (32-shot)	2.910	6.410	4.038	8.010	31.472	39.439
Centralized (16-shot)	2.810	5.810	4.033	7.650	30.320	38.164
Local (16-shot)	2.610	5.456	3.916	7.388	22.722	30.256
Collaborative (16-shot)	2.890	6.099	4.021	7.820	30.416	38.218

Table 3: Few-shot test performance of RAG and parametric models (T5-base and Flan-T5-base) on the CRAB benchmark across different training strategies and shot levels. CoRAG (RAG Collaborative) consistently outperforms parametric models. Collaborative training yields more substantial improvements for RAG than for parametric models, with the performance gap widening as the number of training samples decreases.

Model name	Centralized		Local		Collaborative	
	Exact Match ↑	F1 ↑	Exact Match ↑	F1 ↑	Exact Match ↑	F1 ↑
T5-base	1.862	4.986	1.302	3.814	2.057	5.343
Flan-T5-base	3.142	7.069	2.959	6.852	3.736	7.956
RAG	32.735	41.594	28.222	37.219	31.936	41.125

Table 4: Few-shot performance of parametric models and RAG on the CRAB development set. CoRAG (RAG Collaborative) consistently outperforms the parametric models.

For any set of passages P and client i, we define:

- $R_i(P) = P \cap R_i$: The set of passages in P that are relevant to client i.
- $HN_i(P) = P \cap HN_i$: The set of hard negative passages in P for client i.
- $IR_i(P) = P \cap IR_i$: The set of irrelevant passages in P for client i.

The CoRAG Participation Game We define the CoRAG participation game as follows:

Definition G.1 (The CoRAG Participation Game)**.** The CoRAG participation game is a game with N players (clients), where each player $i \in [N]$ chooses an action $a_i \in 0, 1$: not contributing ($a_i = 0$) or contributing ($a_i = 1$) their passage set P_i to the shared store P_{shared}. Given an action profile $a = (a_1, \ldots, a_N)$, player i's payoff is defined as their utility:

$$U_i(a) = f_i(P_i \cup P_{shared}(a)) - f_i(P_i) - c_i a_i. \quad (1)$$

Here, $f_i(P)$ denotes the performance of player i's model when trained using passages P, $c_i > 0$ represents the cost incurred by client i for contributing, and $P_{shared}(a) = \bigcup_{j:a_j=1} P_j$ is the shared store given the action profile a.

We approximate the performance $f_i(P)$ as:

$$f_i(P) \approx \alpha|R_i(P)| - \beta|HN_i(P)| + \gamma|IR_i(P)|, \quad (2)$$

where coefficients α, β, and $\gamma > 0$ capture the impact of each passage type on performance, with $\alpha > \gamma > \beta$.

Definition G.2 (Nash Equilibria in the CoRAG Game)**.** An action profile $a^* = (a_1^*, \ldots, a_N^*)$ is a pure strategy Nash equilibrium of the CoRAG participation game if, for each player $i \in [N]$ and every action $a_i \in \{0, 1\}$, $U_i(a_i^*, a_{-i}^*) \geq U_i(a_i, a_{-i}^*)$.

Analysis of Client Participation For a given action profile a, define:

- $C(a) = \{j \in [N] : a_j = 1\}$: The set of participating clients.
- $P_{shared}(a) = \bigcup_{j \in C(a)} P_j$: The shared store given action profile a.

A client i participates in a Nash equilibrium a^* if and only if:

$$\begin{aligned} & U_i(1, a_{-i}^*) \geq U_i(0, a_{-i}^*) \\ \iff & f_i(P_i \cup P_{shared}(a^*)) - f_i(P_i) \geq c_i \end{aligned} \quad (3)$$

Conversely, a client i does not participate in a Nash equilibrium a^* if and only if:

$$\begin{aligned} & U_i(0, a_{-i}^*) > U_i(1, a_{-i}^*) \\ \iff & f_i(P_i \cup P_{shared}(a^*)) - f_i(P_i) < c_i \end{aligned} \quad (4)$$

These conditions show that a client participates only if the performance gain from accessing the shared store exceeds their contribution cost. If the

Passage Store Composition	Test Store Only		Test+Train Store	
	Exact Match ↑	F1 ↑	Exact Match ↑	F1 ↑
100% store	31.111	39.760	29.333	37.249
80% store (relevant + others)	30.222	38.685	28.667	35.525
50% store (relevant + others)	31.111	39.015	29.333	37.034
20% store (relevant + others)	31.778	40.835	28.444	35.647
10% store (relevant + others)	31.111	38.969	30.222	37.503
1% store (relevant + others)	29.333	37.418	30.889	39.233
0% store	23.778	29.689	20.889	26.712
Only relevant	29.111	36.467	28.667	38.597
Only hard neg + irrelevant	25.222	32.046	25.556	32.063
Only relevant + hard neg	25.778	32.093	27.111	33.441
Only relevant + irrelevant	32.667	40.569	30.111	36.969
Only top-1 relevant + irrelevant	31.556	40.890	30.333	37.703

Table 5: Performance comparison of RAG (local) across various training store compositions. We assess the impact on Exact Match and F1 scores at test time, using the local test store (I_{test}) only and the combined test and train stores (I_{test} + I_{train}). Scores are averaged across 8 clients.

Passage Store	Client 1		Client 2		Client 3		Client 4		Client 5		Client 6		Client 7		Client 8	
	EM ↑	F1 ↑	EM ↑	F1 ↑	EM ↑	F1 ↑	EM ↑	F1 ↑	EM ↑	F1 ↑	EM ↑	F1 ↑	EM ↑	F1 ↑	EM↑	F1 ↑
REL	3.778	4.684	6.666	7.470	5.999	6.628	5.111	6.571	2.889	3.656	3.999	3.424	7.555	7.519	6.444	6.451
IRR	2.445	4.812	6.000	6.562	6.222	7.427	2.889	4.671	2.000	4.476	5.778	5.895	4.889	6.466	5.778	6.866
REL-1	2.667	4.459	8.444	9.465	3.333	4.018	4.222	4.786	5.334	6.104	5.555	6.261	5.778	5.515	1.445	0.943
SPLIT	4.222	5.248	6.222	7.045	7.112	6.315	6.445	6.063	11.111	11.244	10.000	9.460	7.556	5.700	5.111	5.182

Table 6: Client-specific performance gains (EM and F1) of CoRAG over RAG (Local) for various passage store configurations in the CRAB benchmark.

performance gain is less than the cost, the client will choose not to participate and will only use their local passages.

Using our performance approximation, we can expand the participation condition:

$$\begin{aligned} &\alpha|R_i(P_{shared}(a^*) \setminus P_i)| \\ &-\beta|HN_i(P_{shared}(a^*) \setminus P_i)| \\ &+\gamma|IR_i(P_{shared}(a^*) \setminus P_i)| \geq c_i \end{aligned} \tag{5}$$

The benefit of participation depends on the composition of the shared store relative to the client's local passages. Clients must weigh the potential gain from new relevant passages against the risk of incorporating hard negatives and the impact of irrelevant passages. Clients with many unique relevant passages may be less inclined to participate to maintain their competitive advantage. The equilibrium behavior of clients in this game depends on the distribution of passage types across clients and the individual participation costs.

Mechanisms for Encouraging Participation To address the tension between individual utility and contributing to the collective knowledge base, we propose the following mechanisms:

1. Contribution-Based Rewards: We introduce a reward function that incentivizes clients to contribute high-quality passages:

Definition G.3 (Reward Allocation Mechanism)**.** For a given action profile a, let $C(a) = \{j \in [N] : a_j = 1\}$ be the set of participating clients. The reward for client i is:

$$r_i(a) = \begin{cases} \rho \cdot (|R_i \cap P_i| + \gamma |IR_i \cap P_i|) \cdot |C(a) \setminus \{i\}|, \\ \qquad\qquad\qquad\qquad \text{if } a_i = 1 \\ 0, \quad \text{if } a_i = 0 \end{cases} \tag{6}$$

where $\rho > 0$ is a scaling factor.

This mechanism rewards participating clients based on the quality of their contributions (relevant and irrelevant passages) and the number of other participating clients. The inclusion of irrelevant passages in the reward calculation reflects their value in improving retrieval performance.

2. Tiered Access Levels: We implement a tiered access system based on the quality and quantity of a client's contributions:

$$access_i = \min(1, \frac{|P_i|}{k \cdot \text{avg}_{j \in C(a)} |P_j|}) \tag{7}$$

where $k > 0$ is a parameter controlling the strictness of the access policy. This mechanism provides clients who contribute more passages with broader access to the shared store, incentivizing larger contributions.

3. Reputation Systems: We establish a reputation system that tracks clients' contribution history:

$$reputation_i = \frac{|R_i \cap P_i| - \beta |HN_i \cap P_i|}{|P_i|} \tag{8}$$

This reputation score balances the proportion of relevant passages a client contributes against the proportion of hard negatives, weighted by β to reflect their relative impact on model performance.

CoRAG Game with Incentive Mechanisms Incorporating these mechanisms, we define a modified CoRAG game:

Definition G.4 (CoRAG Game with Incentive Mechanisms)**.** The modified CoRAG game with incentive mechanisms is defined as in Definition G.1, but with player i's payoff defined as:

$$\widetilde{U}_i(a) = U_i(a) + r_i(a) + v_i(access_i) + w_i(reputation_i), \tag{9}$$

where $r_i(a)$ is the reward from Definition G.3, $v_i(\cdot)$ and $w_i(\cdot)$ are non-decreasing functions representing the value player i assigns to their access level and reputation, respectively.

The contribution-based reward encourages participation by compensating clients for the value they add to the shared store. Tiered access levels provide an additional incentive for clients to contribute more passages, while the reputation system introduces a long-term incentive for consistent, high-quality contributions.

This formalization provides a foundation for understanding the strategic considerations of clients in CoRAG and for designing effective incentive structures. Future work could focus on empirically evaluating these mechanisms and analyzing their impact on the Nash equilibria of the modified game.